Journal of Physics: Conference Series **3123** (2025) 012022 doi:10.1088/1742-6596/3123/1/012022

Safety Assessment of Maritime Autonomous Surface Ships: A Scenario-Based Approach

Georg Hake*, Anna Austel, Jan Steffen Becker, Lina Putze and Nina Wetzig

Institute of Systems Engineering for Future Mobility,
German Aerospace Center (DLR), Oldenburg, Germany

*E-mail: georg.hake@dlr.de

Abstract.

Maritime Autonomous Surface Ships (MASS) promise enhanced efficiency in shipping operations, but ensuring their safety presents significant challenges. Traditional distance-based testing, where safety is assumed after travelling a predetermined distance without incident, is impractical for MASS due to prohibitively large distance requirements. Drawing inspiration from the automotive domain, we apply scenario-based testing, where the space of operating conditions is divided into traffic scenarios. Scenario-based testing makes it possible to increase the efficiency of testing by specifically targeting high-risk scenarios. We employ maritime Traffic Sequence Charts (mTSCs) to formally specify test scenarios and corresponding requirements. Our approach encompasses three key elements. First, in a hazard analysis, risk-triggering scenario properties are systematically identified through expert-guided brainstorming and investigation of causal relationship. This results in abstract test cases and safety requirements, which are formally specified as mTSCs. In a second step, concrete test scenarios are generated for each abstract test case by converting mTSCs into mathematical SMT problems and solving these for vessel movements modelled as Bézier splines. Finally, runtime monitors derived from mTSCs are used to continuously evaluate requirement satisfaction during testing. We find that systematic hazard analysis, automated scenario generation, and runtime monitoring can successfully be applied to the verification of maritime systems. The formal specification enables automatic test execution and evaluation in both simulation and real-world environments.

1 Introduction

By taking over steering and control functions, Maritime Autonomous Surface Ships (MASS) have the potential to increase the safety and efficiency of shipping and port operation. However, ensuring the safe operation of autonomous ships in complex open environments remains a significant challenge. Safety concerns as well as difficulties with approval and classification represent some of the primary market barriers for the adoption of autonomous ships within the maritime industry. Stakeholders such as shipping companies, classification societies and port operators consequently have a vested interest in suitable methods for ensuring safety of MASS. In particular, the IMO demands that autonomous ships demonstrate their safe operation in testing [1].

In a classical, distance-based approach to verification and validation, the System under Test (SuT) is assumed to be sufficiently safe after a predefined distance travelled without incidents. This approach is infeasible for MASS, as the distance needed to provide a suitable safety argument is prohibitively large.

For the automotive domain, scenario-based testing has been proposed as a viable solution to this problem [2].

In scenario-based testing, the space of potential operating conditions of the SuT is divided into a collection of traffic scenarios. The idea is to specifically target scenarios associated with a high risk in testing. While a MASS may spend a significant portion of its operating time on the open sea with no other vessel close by or moored in a port, allocating an equivalent proportion of testing resources to these low-risk conditions would be inefficient.

In this paper, we present our research on a systematic scenario-based approach for the safety assessment of MASS, employing maritime Traffic Sequence Charts (mTSCs) [3] for the formal specification of test scenarios and scenario-based requirements. The modelling language of mTSCs is a maritime-specific extension of regular Traffic Sequence Charts (TSCs) [4, 5].

Our focus is on three aspects that are particularly important for this approach: The identification of relevant test cases and associated requirements through a hazard analysis, the generation of concrete test scenarios from the test cases, and the evaluation of test runs with respect to requirements using runtime monitoring. In the hazard analysis, risk-triggering scenario properties are systematically identified and analysed by an expert-driven brainstorming approach and extensive investigation of causal relationships. Based on this, relevant abstract test scenarios and corresponding safety requirements are identified and formally specified using mTSCs. To "play out" the abstract test cases, i.e., generate concrete test scenarios, the mTSCs are converted into a mathematical SMT (Satisfiability Modulo Theories) problem. The movements of involved vessels (modelled as Bézier splines) are generated by finding a solution of the SMT problem. This approach is based on a simplified model of ship dynamics. The generated trajectories are then executed with a simulation software or recreated in a physical test-field. Physical test runs in particular cannot always be guaranteed to match the original specification. To ensure correct test conduction and to evaluate test runs for satisfaction of scenario-based requirements, we employ runtime monitors generated from the corresponding mTSCs. These monitors continuously analyse driving data (position, speed, etc.) and provide information about satisfaction or violation of the specification.

We find that systematic hazard analysis, automated scenario generation, and runtime monitoring can be successfully applied to maritime systems. In particular, the formal specification of test scenarios and requirements enables automatic test execution and evaluation, both in simulation and on real ships. Combined application of these techniques enables the efficient and comprehensive validation of MASS through scenario-based testing.

The paper is structured as follows. In Section 2, we present related research and point out similarities and differences to our work. In Section 3, we discuss the role of abstract traffic scenarios in our testing method. We then present individual steps of the method in Sections 4, 5, and 6. Finally, we conclude with Section 7.

2 Related Work

In this work we present an integrated scenario-based testing approach, combining a hazard analysis for test case selection with scenario formalization, generation, and monitoring methods.

For traditional maritime systems, there are several well-established methods for hazard and risk analysis. Zhou et al. provide a comparative evaluation to assess the applicability of these methods to autonomous vessels [6]. They outline specific strengths of the methods such as Failure Mode and Effects Analysis (FMEA), Fault Tree Analysis (FTA), System-Theoretic Process Analysis (STPA) and Hazard and Operability Study (HAZOP), noting that most of these methods generally require adaptation or integration with complementary methods to effectively address the challenges posed by autonomous maritime systems. Li et al. provide a literature review focusing on risk and reliability research specifically targeting MASS [7]. They report on different combined approaches, such as the integration of FMEA with Bayesian networks and evidence reasoning [8] and an ontology-based FTA [9], as well as several applications of STPA. For instance, Rokseth et al. compare STPA with FMEA [10], while Banda et al. integrate STPA into a broader safety management framework applied to autonomous ferries. Further, Pedersen et al. extend the application of STPA towards simulation-based testing by formalizing the STPA results using Signal Temporal Logic (STL) [11, 12]. They combine their approach with a guided testing method based on Gaussian process models proposed by Torben et al. [13]. The use of STL allows the authors to apply robustness metrics in the test evaluation, i.e., to quantify how well the system satisfies a test requirement and give a confidence for untested cases. In contrast, our approach uses abstract test scenarios, in the form of mTSCs, for test case generation and evaluation. Test scenarios generated from mTSCs are not limited to a fixed parameter space (knowing the set of test parameters is a prerequisite for the approach by Torben et al.) and therefore allow a greater diversity. Furthermore, mTSCs are specifically designed for the specification of complex spatial relationships which are cumbersome to formalise in STL.

To the best of our knowledge, there is currently no standardised and formal scenario description language available in the maritime domain for modelling and maintaining abstract traffic scenarios. However, there have been several attempts to formalise maritime traffic rules [14, 15, 16, 17]. In their literature review, Porres et al. note that the majority of research papers select a small number of standard scenarios, such as crossings or takeovers, and specify them manually [18]. To counteract this, Pedersen et al. propose a series of standard ship encounters that can be used to generate variants of traffic situations to test compliance with various COLREG rules [12]. In another work, Porres et al. present a method for scenario-based testing focusing on an algorithm for scenario selection [19]. In this context, a scenario is defined by the starting positions, following way points, velocities and characteristics of the vessels involved. In contrast to this work, the authors consider vessel characteristics and dynamics, without accounting for environmental factors.

In the maritime domain, scenario descriptions are more commonly applied in anomaly detection. Different methods are used to formalise the scenarios to search for possible deviations. Riveiro et al. categorise grid-based, vector-based and graph-based methods for representation of behaviour [20].

To evaluate the compliance of autonomous ships with scenario specifications, there are different approaches to monitor and evaluate a ship's behaviour. Lourenco et al. introduce a monitor based on STL specification of traffic rules in the automotive domain [21]. The monitor described in their work uses a complete system trace and therefore cannot be used at runtime. Goyal et al. describe a monitor based on Linear Temporal Logic (LTL) that is used at runtime [22]. Their monitor is part of a framework for simulative testing of autonomous driving systems in the automotive domain. For the maritime domain, Tan and Tng developed an approach for the specification of scenarios and the monitoring of driving systems [23]. Their monitors can be used in post processing and is not suitable for runtime monitoring.

To test and demonstrate safe operation of MASS for autonomous ships with an autonomy level of three or four as specified by the IMO [24], Perera proposes a three-level approach to scenario-based verification of MASS, from simulated over hybrid models to full-scale vessels in a test-bed [25]. Current scenario- and simulation-based verification approaches applied in the maritime context predominantly use co-simulation frameworks, bridge simulators and stand-alone simulation platforms [26]. One example is the Open Simulation Platform (OSP) [27], which allows different manufacturers to integrate their individual components in a co-simulation framework, which can be hosted by a system integrator, yard or by classification societies.

3 Abstract Traffic Scenarios in Testing MASS

In this section we discuss the role of abstract traffic scenarios and formal specifications thereof in the verification of highly automated and autonomous ships.

Consider as SuT a ship about 40 meters in length, with route-following and collision avoidance capabilities. Among other safety requirements, this system will be required to operate in compliance with the Convention on the International Regulations for Preventing Collisions at Sea (COLREG) [28]. We use this SuT as a running example throughout this paper.

A ship like our SuT is likely to spend a substantial portion of its operation in low-risk conditions, performing route-following tasks in areas with low traffic and in mild weather. However, it will inevitably encounter some high-risk conditions in which it has to deal with complex circumstances, e.g., while navigating through ports or other areas with high traffic. To avoid wasting effort during testing, our approach aims at targeting operating conditions associated with an increased risk.

Behavioural requirements for ships do not only depend on current circumstances but also on preceding events. For instance, according to COLREG Rule 7(d)(i), a risk of collision is to be assumed whenever the compass bearing of an approaching vessel does not significantly change for some time. According to COLREG Rule 13(d), it is not possible to classify an encounter of two vessels as overtaking or crossing based solely on the current bearing between them, as it also depends on how that bearing evolved previously. Consequently, to plan their next action in compliance with existing rules and regulations, automated vessels have to take into account not only the current state of their environment, but also the previous temporal evolution of traffic conditions. This means the input space of a highly automated ship can be considered to be the space of all *traffic scenarios* it might encounter, i.e., the space of all possible evolutions of traffic and environmental conditions around the ship over time. This motivates the use of scenario-based methods, where the SuT is confronted with selected traffic scenarios representative of its input space. The focus is on critical traffic scenarios to see whether the ship adheres to its requirements even under difficult conditions.

Specifically, the space of possible operating scenarios is subdivided into categories of roughly similar circumstances, defined by abstract traffic scenarios, cf. [29]. Within these basic categories, critical variants with some kind of risk-triggering scenario properties are identified. This includes properties that might

make accidents more likely as well as those under which accidents may have particularly catastrophic consequences. Identified hazardous scenarios (also described as abstract traffic scenarios) then serve as instructions on what conditions the ship should be tested in. Based on an analysis of how safety requirements and traffic regulations apply in each abstract test scenario, corresponding requirements on the SuT's behaviour for each test case are derived. These requirements in many cases involve multiple subsequent manoeuvres and actions and can be specified as abstract traffic scenarios as well. Tests can then be conducted in a traffic simulation or test-field by observing system behaviour under the conditions of each test scenario and evaluating it against applicable requirements.

To provide objectivity and enable automation of this testing procedure, basic scenarios, their critical variants, and the corresponding requirements need to be formally specified. In this work, we use mTSCs for this purpose. MTSCs are a visual formalism for the specification of abstract maritime traffic scenarios. They describe traffic scenarios in terms of phases with invariant conditions arranged in a *chart* representing the temporal and logical structure of the scenario. Each phase is characterised by a so-called *spatial view*. Spatial views provide visual formalization of abstract traffic situations with a focus on intuitive representation of involved objects (like ships, port infrastructure, etc.) and their spatial properties and relations (like positions, directions, distances and angles). The intuitive visual representation of scenarios makes mTSCs human-readable, even without a background in formal methods. This has the advantage that different stakeholders can be involved in the specification process and mTSCs can be used to provide instructions to personnel conducting field-tests. MTSCs have previously been used for the specification of maritime traffic scenarios [3, 30]. For a more detailed account of the TSC language refer to [5].

Based on a formal specification of test scenarios and corresponding requirements as mTSCs, the testing process can be partially automated. For simulation-based tests, concrete scenarios representative of each abstract test scenario can be automatically generated. Generated routes can also be the basis for conducting field-tests. For a given mTSC a monitor can be created to automatically assess concrete scenarios with respect to the specification. Such monitors may be used to automate test evaluation and check test conduction against the specification of the test scenario.

In the following sections we explore three aspects of a scenario-based testing process for highly automated and autonomous ships in detail: The identification of critical scenarios, the creation of concrete scenarios to test the ships in and the evaluation of test runs with respect to corresponding requirements.

4 Derivation of Abstract Test Cases

The basic idea of the scenario-based approach for the verification and validation of automated maritime systems is to structure the system's target operational domain using scenario classes. The selection of relevant scenarios and corresponding requirements then forms the basis for the subsequent test process.

Scenarios associated with an increased risk are considered to be particularly relevant. To identify such scenarios, hazards known from previous experience with similar systems can be used. These can be determined on the basis of both data and expert knowledge, e.g. using criticality analysis for the corresponding system class [31]. In addition, system-specific hazards have to be taken into account. Therefore, the selection of relevant test scenarios necessitates a systematic process to identify and analyse potential hazards and risks of the system under development. For automated systems, solely ensuring functional safety, as typically emphasised in conventional maritime safety analyses, is insufficient. It has to be ensured that the system specification is inherently safe and does not give rise to hazards resulting from insufficiencies of the specified functionality, such as gaps in the specification or limited system performance.

To detect such hazards, we rely on a method for identifying and analysing hazardous scenarios originally proposed by Kramer et al. for the automotive domain [32, 33]. The approach addresses both functional safety and the safety of the intended functionality (SOTIF) by building on well-established methods, while adapting them where necessary to accommodate the specific challenges posed by automated systems. In context of this work, we adapted this method specifically to integrate it into scenario-based safety assessment for MASS and applied it to our running example. The hazard identification process shall accompany the complete system development starting early in the concept phase, so that safety issues can be addressed from the outset. Iterative feedback mechanisms between the concept and development phase enable a systematic identification, evaluation, and implementation of risk mitigation strategies. Figure 1 provides an overview of the different steps of our approach to derive abstract test cases and safety requirements.

First of all, a model of the system needs to be defined. This includes a functional architecture of the system that decomposes the different functionalities, including the flow of information between sensors, perception modules, planning units, and actuators. In addition, a model of the ODD needs to be set up that provides a formalization for the environmental conditions.

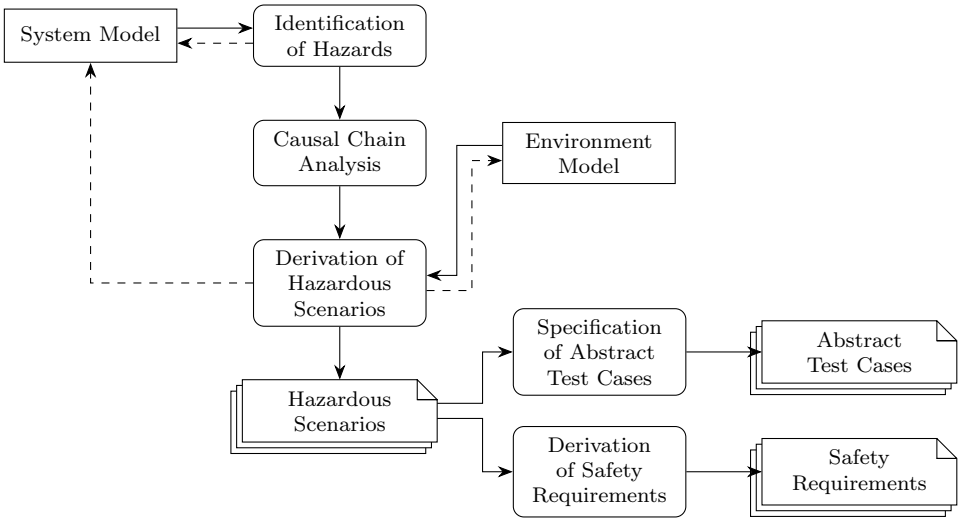

Figure 1: Approach to derive abstract test cases and safety requirements based on the method proposed by Kramer et al. [32].

As starting point for the hazard identification, basic scenarios must be defined, that cover the system's target operational domain. These basic scenarios provide the foundation for the subsequent analysis steps. A set of COLREG-triggering encounters that can be used to generate a set of basic scenarios have been for example proposed by Pedersen et al. [12]. The general idea is to gradually vary and refine the basic scenarios to identify hazardous scenarios. As example, we consider the basic scenario *overtaking vessel*, depicted in Figure 2. The overtaking vessel, also called the *target ship* (`TS`) is assumed to be of comparable size to the vessel under investigation, also called the *own ship* (`OS`), i.e. about 40 meters in length. The initial distance between them is selected accordingly based on expert knowledge. Further, the angles are defined based on the definition of overtaking given by COLREG. For the specification we rely on mTSCs. The formal representation facilitates a common understanding among systems engineers and supports the mapping of abstract test cases to corresponding basic scenarios.

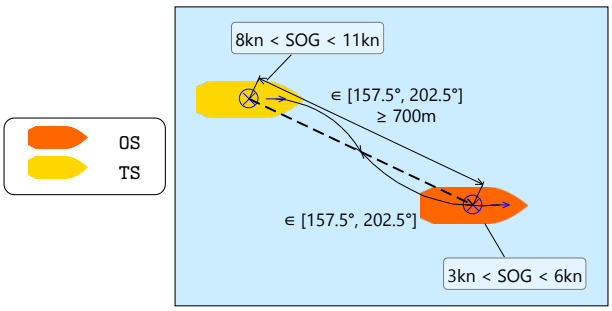

Figure 2: Basic scenario *overtaking vessel* specified as an mTSC.

For the hazard identification two complementary sub-steps are conducted combining bottom-up and top-down approaches: First, hazardous behaviours and hazardous events at vessel level are identified by applying a keyword-based brainstorming approach. Each basic scenario is paired with a set of actions that the vessel under development is capable of, such as *change course*, *maintain course*, *accelerate* or *decelerate*. In Table 1, to provide a specific example, we investigate the combination of the basic scenario *overtaking vessel* from Figure 2 and the basic action *change course*. For each combination of basic manoeuvre and basic action, a context is defined in terms of refined operating conditions, under which the basic action is considered appropriate. Considering the example of Table 1, a course change is appropriate if a last moment manoeuvre is required by COLREG. To derive hazardous behaviours, each combination of basic scenario, basic action, and context is further evaluated by applying a predefined set of keywords, such as *no*, *less*, *more* and *too early*. These keywords are applied either to the manoeuvre

or to conditions of the context to explore deviations from the expected system behaviour. For instance, applying the keyword *less* to the basic action *change course* reveals the hazardous behaviour *insufficient course change* which may lead to a collision with the overtaking vessel. If the application of a keyword uncovers a system behaviour that potentially leads to some harm like in this case, the resulting hazardous behaviour and its downstream consequences are documented. These comprise observable effects in the scenario, the resulting hazardous event, and additional scenario conditions that may contribute to the risk.

Table 1: Table for identification of hazards on vessel level.

ID	Basic Scenario	Basic Action	Correct if (context)	Key-word	(Hazardous) behaviour	Observable Effect(s) in Scenario	Hazardous Event	Additional Scenario Conditions
1	Overtaking vessel	Change course	Collision course and distance $\leq$ last moment manoeuvre distance $\rightarrow$ Change course away from overtaking vessel	no	No course change away from overtaking vessel	Ship (ego) maintains collision course even though a last moment manoeuvre is required	Collision with overtaking vessel	-
2				less	Insufficient course change away from overtaking vessel	Ship (ego) remains on collision course	Collision with overtaking vessel	-

The second sub-step of the hazard identification focuses on detecting local failures and functional insufficiencies that may lead to hazardous behaviours at vessel level. To this end, each functional unit (FU) of the functional architecture is systematically analysed by applying a set of predefined keywords to each function. For example, in Table 2 the keyword *less* is applied to the functional unit *rudder* discovering the failure / functional insufficiency *insufficient rudder deflection*. The resulting deviations are then propagated forward to assess their impacts at vessel level. Each identified local failure or functional insufficiency is evaluated in context of the different basic scenarios to determine whether it may contribute to some hazardous behaviour at vessel level. If such behaviour can be identified it is traced back and cross-referenced with the corresponding entries in Table 1. For instance, in the basic scenario *overtaking vessel* the failure / functional insufficiency *insufficient rudder deflection* may result in the hazardous behaviour *insufficient course change* as recorded in the second line of Table 1. Finally, relevant system causes and environmental triggers are denoted to support a comprehensive understanding of the hazard propagation chain.

Table 2: Table for identification of local failures / functional insufficiencies.

Functional Unit (Input, Computa-tion, Output)	Key-word	Local Failure/ Functional Insufficiency	Basic Scenario	System Effect(s) in Scenario	(Hazardous) behaviour	ID(s) of HB	System Cause(s)	Env. Trigger
Rudder (Control signal, processing, rudder position)	less	The rudder deflection is insufficient	Over-taking vessel	Insufficient course change away from overtaking vessel, ship (ego) remains on collision course	Insufficient course change away from overtaking vessel	2	Hardware malfunction, missing control signal, failures of processing algorithm	-

In the previous step, including its sub-steps, the focus has been on single faults and failures. To understand how combinations of faults and functional insufficiencies interact with triggering conditions and propagate through the system leading to the identified hazardous events, so called environmental fault trees (EFTs) are employed. Unlike a conventional fault tree analysis, which focuses primarily on the system's hardware, this method models environmental conditions within the fault tree as necessary enablers for fault propagation. Each EFT is rooted at a previously identified hazardous event while the corresponding basic scenario is assumed as context. Relying on the functional architecture, the hazardous event can be decomposed step by step based on the results of the previous step.

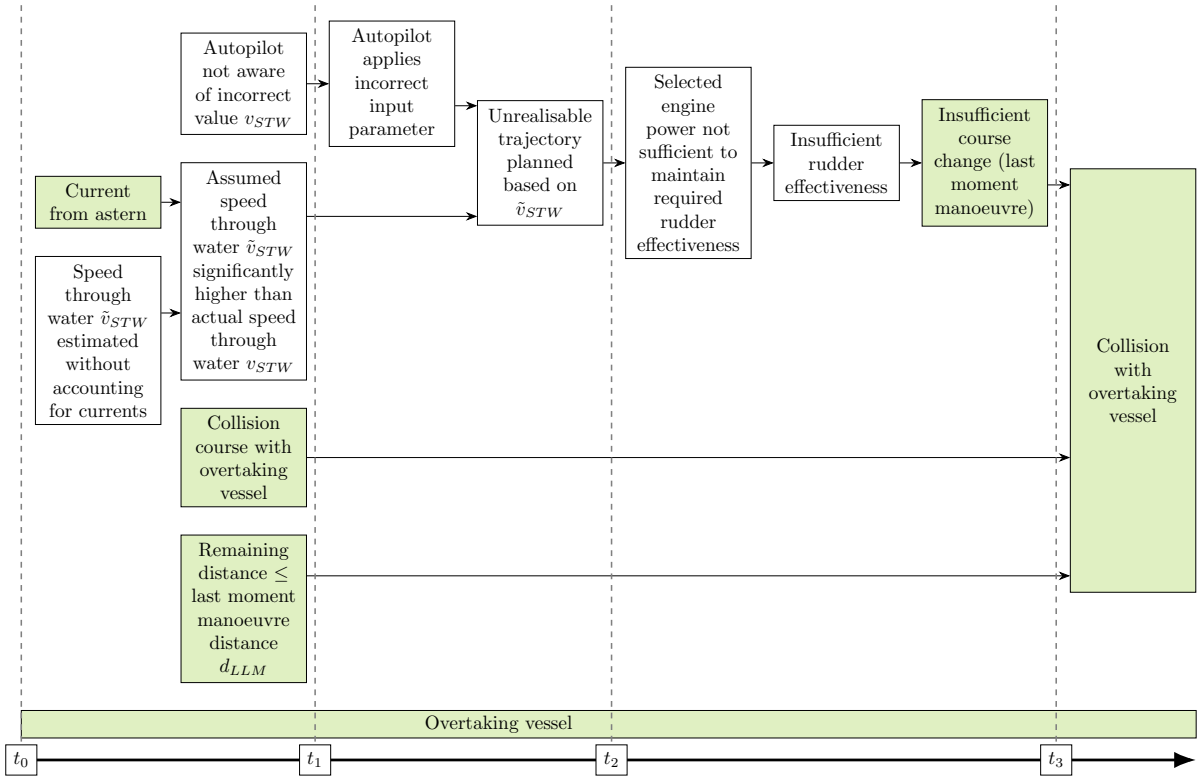

Figure 3: Hazardous scenario including local failures, functional insufficiencies and triggering environmental conditions. Environmental conditions and vessel level information are highlighted in green.

The EFTs serve as foundation for deriving scenario properties that may trigger the identified hazardous events. For the next steps, we deviate from the original method proposed by Kramer et al. [32] placing greater emphasis on causal dependencies within the system, as the understanding of the system's causalities facilitates the derivation of safety requirements and pass/fail criteria for the specified test cases. We begin by analysing each minimal branch set of the EFTs separately. Here minimal branch sets are sets of branches connected via AND-gates that are sufficient to cause a hazardous event. Environmental conditions are formalised using the environment model. To represent the temporal unfolding of events within the minimal branch sets, the events are assigned to discrete time steps, with each time step defining a distinct scene. The relative ordering of these scenes captures the temporal evolution of events. Thus, it provides a scenario description including local failures, functional insufficiencies and triggering conditions. Figure 3 illustrates an example of such a hazardous scenario. It includes the hazardous behaviour *insufficient course change* and the corresponding local failure / functional insufficiency *insufficient rudder deflection* identified in Table 1 and Table 2 as well as further relevant local failures, functional insufficiencies and triggering environmental conditions.

The identified hazardous scenarios enable a targeted definition of relevant test cases at vessel level. We aim at test cases in which it is up to the vessel under investigation to avoid harm by taking appropriate actions. Therefore, each hazardous scenario is abstracted by restricting the hazardous scenario to events containing environmental conditions and vessel level information (both highlighted in green). Based on these scenarios at vessel level, abstract test cases can be derived and specified using mTSCs. The

specification of an abstract test case may require refinement of single events, such as a concretisation of certain conditions or a selection of specific values or value ranges.

Figure 4 illustrates an example of such an abstract test case, derived from the hazardous scenario presented in Figure 3. The mTSC models an encounter with an overtaking vessel (`TS`) in the presence of current. It shall be tested whether the vessel under investigation, the own ship (`OS`), takes appropriate actions to avoid a collision with `TS`, despite the presence of a current from astern that results in a speed through water (STW) of `OS` that is significantly lower than its speed over ground (SOG). To test this, the behaviour of `TS` is specified such that a collision would occur if `OS` were to maintain its initial course and speed, i.e., a constant speed over ground (SOG) of 5 knots and a rate of turn (ROT) of 0 degrees per second. To specify this behaviour of `TS`, the hypothetical behaviour of `OS` is included in the mTSC. However, aside from its initial state, the trajectory of `OS` will be replaced by its simulated behaviour in concrete test scenarios, in order to enable an evaluation of the system's behaviour in testing, cf. Section 5.

The mTSC depicted in Figure 4 specifies the behaviour of `TS` relative to the hypothetical behaviour of `OS` as follows: At the beginning of the scenario, `TS` has an initial SOG of 10 knots, a ROT of $0°/s$ and is on course to collide with `OS` after 6 minutes. Subsequently `TS` maintains a SOG greater than 10 knots and a ROT less than $0°/s$, i.e., the overtaking vessel initiates a course change away from `OS`. Despite this behaviour, the scenario ends in a collision with a relative angle of the courses of both vessels ranging between $-1°$ and $7°$, i.e., the course change in the second phase is insufficient to avoid a collision without further actions of `OS`.

In addition to abstract test cases, safety requirements can be derived for the hazardous scenarios. For example, for the given basic scenario of Figure 2, a requirement on vessel level such as *the vessel must avoid a collision and adhere to the COLREG rules in the process*, may be formulated. These high level safety requirements can be further decomposed and allocated to specific system functionalities by leveraging the causal chains of the identified hazardous scenarios. The resulting requirements can then be formally specified as pass/fail criteria for corresponding test cases, and monitors can be constructed for automatic test evaluation, cf. Section 6.

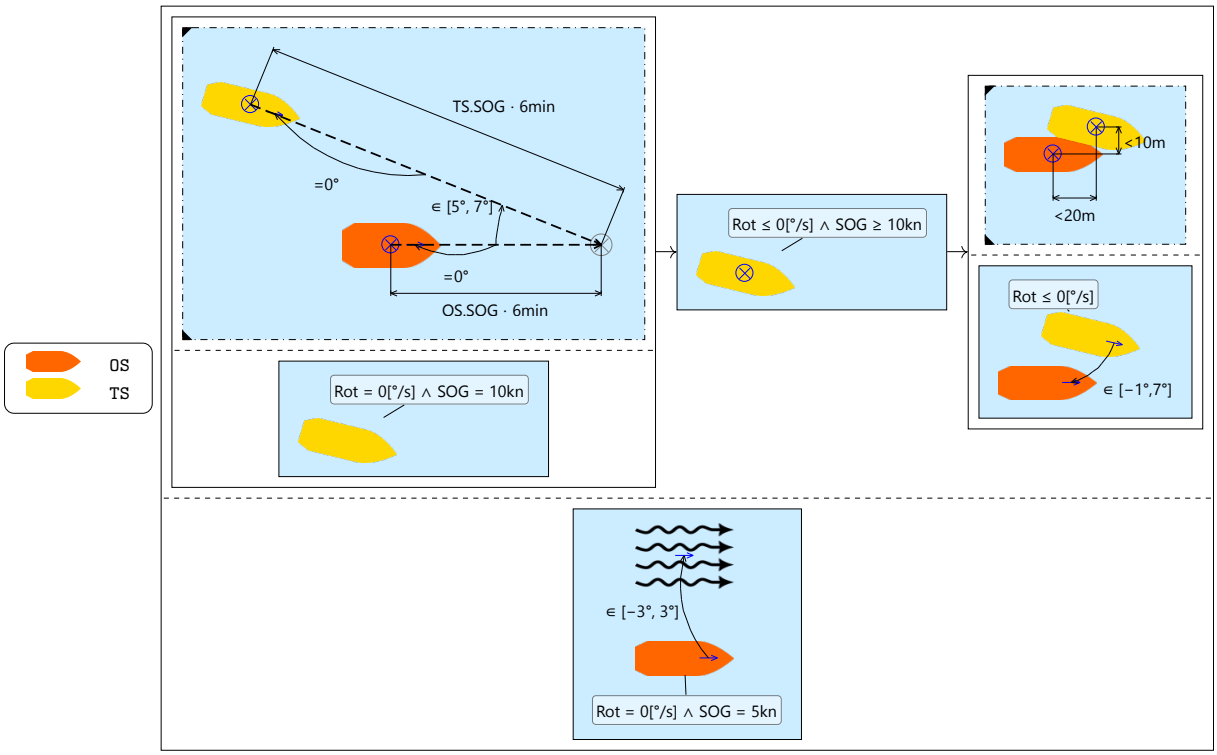

Figure 4: Abstract test scenario modelling an encounter with an overtaking vessel in the presence of current specified as an mTSC.

5 Creation of Concrete Test Scenarios

In order to be able to simulate and evaluate the abstract scenarios created as described in Section 4, they must first be instantiated into concrete scenarios. Traditionally, especially in the automotive industry, traffic scenarios are simulated by modelling them as a sequence of predefined manoeuvres, each of which is parametrised according to the scenario. This procedure is not applicable to abstract scenarios, as only the relations between the participants and their environment are specified instead of the manoeuvres. This results in more diverse scenarios, but at the same time more complex dependencies than can be modelled using a classical approach. We therefore use an alternative approach in which the generation of concrete scenarios is based on the solution of complex linear mathematical problems. The problem class used is referred to as *satisfiability modulo theories* (SMT) [34]. There are specialised programs optimised to solve these problems, so-called SMT solvers.

The generation of specific test data then consists of the following steps:

1. Modelling the abstract scenario as mTSC
2. Converting the modelled mTSC into an SMT problem
3. Enumerate solutions of the SMT problem
4. Translate the solutions into simulation files
5. Simulation of the scenarios generated in this way

The trajectory of the SuT is then removed and replaced by simulating its behaviour in the generated scenario. The resulting behaviour is evaluated against requirements as described in Section 6.

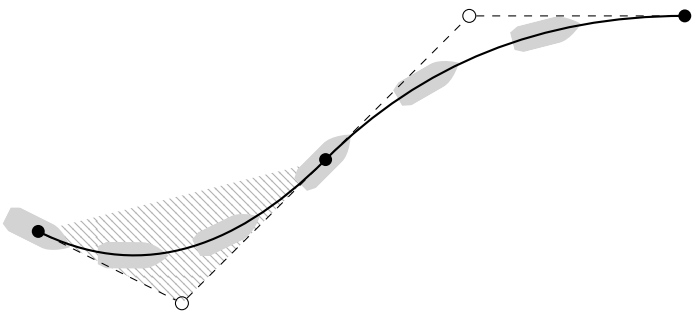

Figure 5: A quadratic Bézier spline trajectory consisting of two segments. The first segment's control point polygon is shaded.

A method that is already described in the literature [35, 36] is used for the translation into an SMT problem. The main idea of this method is to model the trajectories of traffic participants (here: ships) as so-called Bézier splines (Figure 5). A Bézier spline can be uniquely described by a few control points, which is why they can be coded in an SMT problem.

The method takes into account vessel characteristics such as maximum speed, lateral and longitudinal acceleration limits and minimum curve radius. The generated trajectories describe the movement of a fixed reference point of the vehicle. It is assumed that the orientation of the vehicle is tangential to the trajectory at the reference point. This means that the reference point is usually shifted relative to the centre of gravity of the vehicle on the longitudinal axis of the vehicle. For road vehicles (for which this model was developed), the reference point is therefore on the non-steered rear axle. In relation to ships, this point is referred to as *pivot point* [37]. To simplify the trajectory generation, it is assumed that the position of the pivot point in relation to the ship is known in advance and is constant during the scenario. The reader may notice that this assumption simplifies real vessel dynamics may not hold for all scenarios. Also it does not account for influencing environment factors such as current and wind which usually cause a divergence between heading and curse. However, since the exact ship dynamics are taken into account by the simulator when executing the scenarios, no mathematically correct coding of the ship dynamics in the SMT problem is required. It is only necessary that the generated trajectories allow for a sufficiently accurate navigation. Furthermore, the techniques applied in Section 6 for test evaluation can be used to monitor whether the simulated vessels behave as specified in the abstract test case. Hence, infeasible trajectories can be detected and discarded.

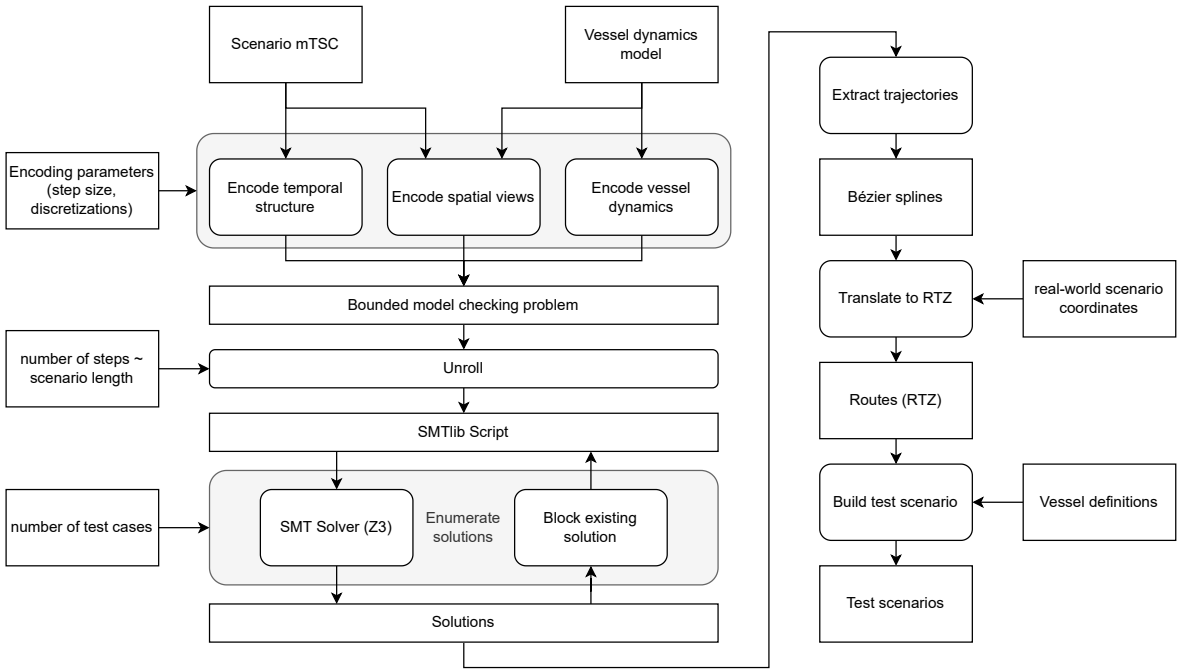

Figure 6: Process for scenario generation

The different steps of the scenario generation are summarised in Figure 6. The temporal structure of the mTSC, the spatial views, and the vehicle dynamics are encoded separately. The result of the encoding process is a bounded model checking (BMC) problem with a fixed step size Δ, which is the time progress in each step. An SMT problem for scenarios of length $N \cdot \Delta$ can be produced by unrolling the BMC problem for N steps. The vessel movements are described using one Bézier spline segment per step. The spatial views are encoded in a way that they hold among all control points of the segment. Relations which cannot be expressed linearly in terms of Bézier control points (such as a course difference which would require non-linear arithmetic), are safely approximated by linear upper and lower bounds, using user-provided discretizations. Because Bézier spline segments never exceed the convex polygon described by their control points, this technique ensures that a spatial view holds during the complete duration of the step, and not only at its beginning (as it is the case for classical discretizing approaches). As a consequence, big step sizes (e.g., 30 seconds) are possible without infringing the validity of the generated concrete scenario with respect to the TSC.

An SMT solver can generate at least one solution from any solvable SMT problem. Any number of additional solutions can be enumerated using various techniques (e.g. atom blocking [38]). Trajectories can be extracted from each of these solutions and translated into simulation data. Figure 7 shows the trajectories for the own ship (`OS`) and the target ship (`TS`) from the first solution returned by the solver for our example mTSC in Figure 4. The instance has been generated with a step size of 20 seconds and unrolled for 18 steps, yielding a 6 minutes long scenario. Figure 7 shows the vessel positions after some of the steps, corresponding to $t = 0, 3, 4, 5, 6$ minutes. Note, that the solver does not avoid collisions. This is as intended in our case, as we explicitly model a collision scenario and want colliding routes as test data. If needed, however, additional constraints can be added to the SMT problem that force the solver to search for collision-free scenarios only.

As an intermediate representation, we store the trajectories as route information in RTZ format [39]. This route information is then processed by the maritime traffic simulation MTS [40] from the virtual test-bed of the eMaritime Integrated Reference Platform [41]. The translation of the SMT solutions into simulation files largely consists of translating the trajectory into RTZ. Because the Bézier control points are represented as free variables in the SMT problem, the positions of the control points can easily be extracted from the solution. Using Bézier spline interpolation, we can calculate as many way points as needed for the simulator to precisely follow the trajectory, plus the required speed in each point. For our example scenario, we generate a waypoint every 10 seconds. More dense way points represent the intended trajectory more accurately, but are harder to follow by the controller that steers the vessel in

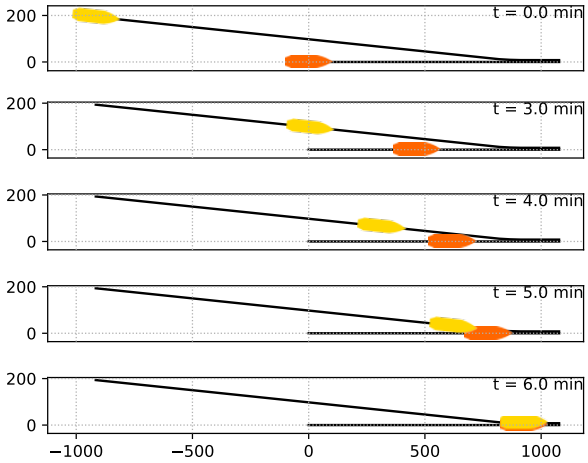

Figure 7: An example instance of the TSC in Figure 4 with the positions of `OS` and `TS` in the beginning ($t = 0$) and after $t = 3, 4, 5, 6$ minutes. Axis labels represent the Cartesian coordinate system used for encoding the TSC. For better readability, the vessels are drawn larger than their natural size.

the simulation—larger distances between waypoints yield a more robust simulation. As the MTS works on real maps, but the trajectories are generated in a local Cartesian coordinate system, the trajectories must be projected into a WGS'84 coordinate system. For this purpose, a reference point is selected in the generated concrete scenario (e.g. the initial position of the ego-ship or a fixed surrounding object, such as a quay wall), as well as a corresponding point on the map. The direction of the X axis in the scenario is also defined. This then clearly defines the coordinate transformation that is used to translate waypoints from the trajectories into RTZ. The last part is the provision of a ship model and bundling the data into a simulation setup.

6 Test Evaluation for Scenario-Based Requirements

To automatically evaluate the behaviour of an automated vessel with respect to applicable requirements we propose to use online monitors constructed based on a formalisation of requirements using mTSCs. In this section we present our approach to using monitoring for test evaluation and present an example of such a monitor for a requirement corresponding to the example test case discussed in previous sections.

For the critical scenario specified in Figure 4, we derived a requirement by applying appropriate rules from COLREG. Faced with a target ship (`TS`) approaching on a collision course from astern, the own ship (`OS`) is initially required to keep its course and speed over ground. Once it becomes apparent that the evasive action taken by `TS` is insufficient to avoid a collision by itself, `OS` is then required to act in a way to best avoid a collision. In this case that means the own ship may increase its speed and turn right, away from the target ship's course. The overtaking scenario is considered finished once `TS` has passed `OS`.

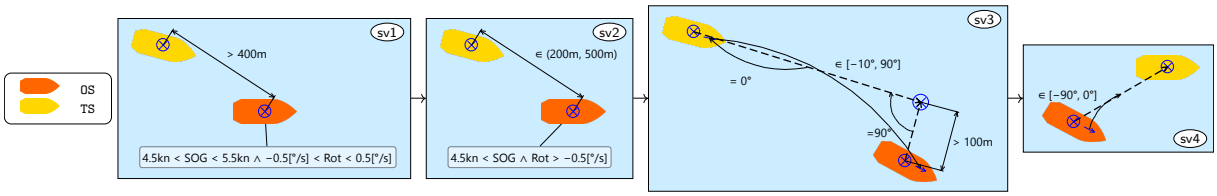

Figure 8: mTSC specifying required behaviour for the own ship, given that the target ship behaves as specified by the critical scenario from Figure 4.

We can formalise this requirement with an mTSC for example as shown in Figure 8. Exact values depend on the specific setting (e.g. ship size, turning circle). The mTSC specifies the requirement as a sequence of four subsequent phases.

Initially, the own ship has to hold its speed over ground and course approximately constant. This is specified as a speed over ground (SOG) close to 5 knots (between 4.5 and 5.5 knots specifically) and a rate of turn (ROT) close to 0 (between −0.5 and 0.5 degrees per second) in the first spatial view.

Following this, starting at a distance between 400 and 500 meters, the ship may take evasive actions by increasing its SOG and ROT, i.e., turning right. This is specified by the second spatial view.

As indicated by the third spatial view it must do so in a way as to achieve a distance of at least 100 meters to the projected course of `TS`, to guarantee a safe distance for passing. This has to happen before the distance between the two vessels falls below 200 meters, as specified by the minimum distance allowed in the second spatial view.

It must then maintain this distance to the projected course and may not approach it at an angle steeper than 10 degrees until the overtaking vessel has moved past the ship, i.e., reaches a relative bearing between −90 and 0 degrees (front left of `OS`), as indicated in the final spatial view. At this point the collision has been avoided and both vessels may return to their planned routes.

Given this mTSC we can construct an online monitor for the requirement following the method described by Stemmer et al. [42] for classical TSCs and adapted to the maritime domain by Austel et al. [3, 30]. This monitor can then be used for automated test evaluation.

Online monitoring of a scenario here refers to the process of evaluating at runtime whether a given drive can still satisfy the mTSC or not. More specifically, at regular intervals during the drive, the monitor receives data for relevant attributes of each involved vessel (speed, rate of turn, relative position, etc.) referenced in the mTSC. Based on this data, the monitor decides which, if any, spatial views are currently satisfied. From the satisfaction of spatial views over time the monitor determines whether the overall mTSC is already satisfied, can still be satisfied, or will be violated no matter what happens next. It continuously provides corresponding verdicts; *satisfied*, *inconclusive* and *violated* respectively. The technical implementation of the runtime monitoring essentially consists of three components: Data collection from simulation (or vessels involved in a field test), assessment of satisfaction of individual spatial views and the computation of a verdict based on spatial view satisfaction over time.

To extend the example from the previous sections to test evaluation using online monitors, we apply our online monitor to two simulation runs created on the basis of the generated concrete scenario presented in Figure 7. Starting from this concrete scenario running in the MTS, we simulated different possible behaviours of the own ship in this setting. Here, the target vessel's behaviour was generated from the critical test scenario as described before. The routes depicted are planned routes for both vessels.

We then monitor the simulation runs for satisfaction of the requirement as specified by the mTSC from Figure 8. We present two of the scenarios below, to illustrate the application of mTSC monitors in test evaluation. Note that we do not monitor the planned routes but rather the vessels' simulated behaviour which might deviate from the planned routes depending on ship dynamics, and additionally include each vessel's speed and other attributes beyond their position.

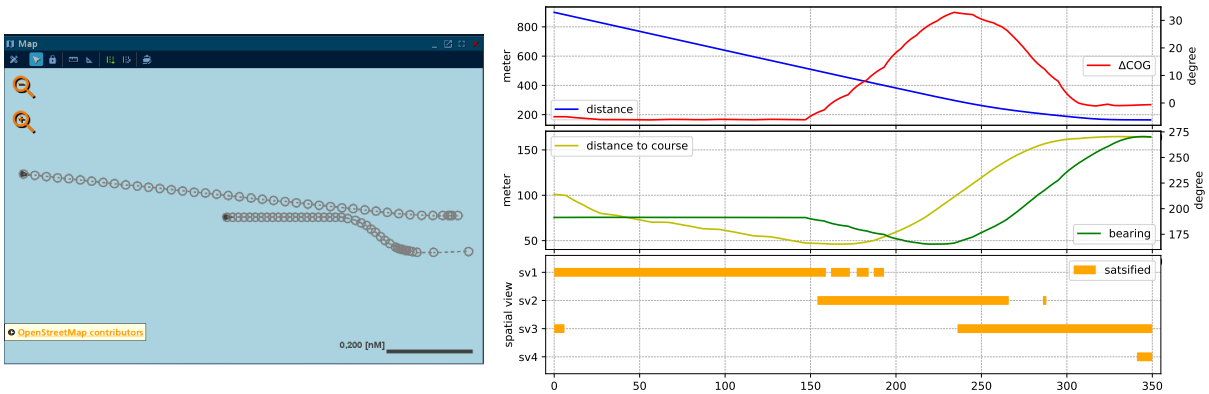

Figure 9: Overtaking scenario in the simulation MTS with target vessel following trajectory generated from mTSC test scenario specification (left) and selection of relevant data and satisfaction of spatial views over the course of this scenario, as determined by our monitor (right). Here *distance* is the distance between both vessel's positions, ΔCOG is the angle between the target and own vessel's courses, *distance to course* is the distance of the own vessel's position to the projected course of the target vessel and *bearing* is the relative bearing of the target vessel from the own vessel.

As a first example we present a simulation run in which the requirement is satisfied. Figure 9 shows a screenshot of our example scenario running in the MTS and corresponding plots of a selection of relevant data used by the monitor to determine satisfaction over the course of this scenario. The vessel's speed over ground for example is approximately constant throughout the scenario and within the permitted range, so we omitted it in the plots.

The scenario begins with the own vessel keeping its initial speed and course over ground for some time. When the distance between the ships approaches 500 meters, the own ship starts turning right. In the plots this is visible in the increasing angle between the ships courses and relative bearing. The distance falling below 500 meters marks the beginning of the second phase. Still, the first spatial view is intermittently satisfied for some time as the change in course does not consistently exceed 0.5 degrees per second. By turning away, the ship manages to build some distance to the overtaking vessel's projected course. Eventually, this distance exceeds 100 meters and the third spatial view is satisfied. Roughly at this point the ship begins turning back towards its original course and the angle between both ship's courses slowly falls to 0. With the ships at roughly parallel courses at a solid distance of more than 160 meters, the overtaking ship finally passes the own ship and the fourth spatial view is satisfied, marking the successful completion of the scenario. The monitor detects and reports the satisfaction of the mTSC as soon as the final spatial view is first satisfied. At this point the test run could be terminated and considered a pass.

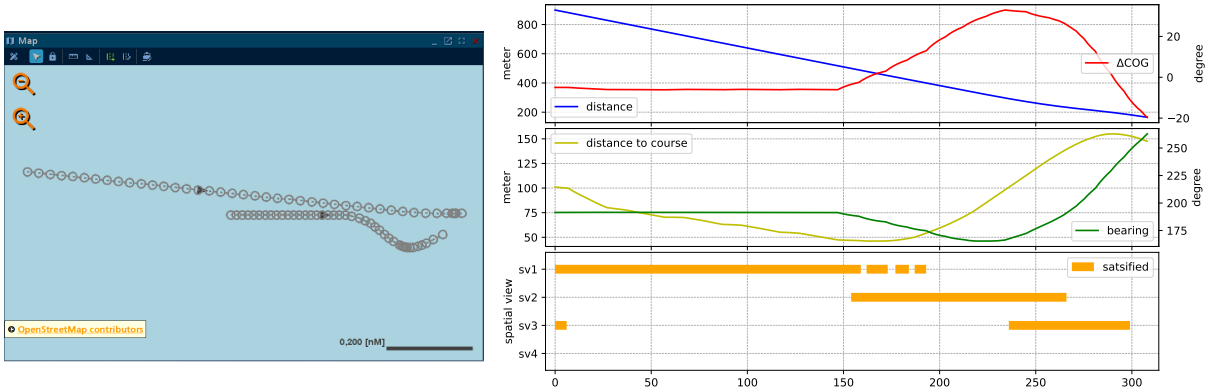

Figure 10: Overtaking scenario in the simulation MTS (left) with target (upper) vessel following trajectory generated from mTSC test scenario specification and own (lower) vessel violating the requirement from Figure 8 by returning to its original course too early. Selection of relevant values and satisfaction of spatial views over the course of a scenario violating the requirement as determined by our monitor (right). Here *distance*, ΔCOG, *distance to course* and *bearing* denote the same properties as in Figure 9.

As a second example we present a scenario in which the requirement is violated. Figure 10 shows a screenshot of the drive in the MTS on the left and plots of the corresponding data as determined by our monitor on the right. Up to the beginning of the third phase this scenario is identical to the successful scenario discussed before. Once the own vessel has reached a good distance to the overtaking vessel's projected course, instead of continuing on a parallel course to `TS`, the own ship attempts to return to its original route immediately, as can be seen in the planned route depicted in Figure 10 on the left. This leads to the minimum required distance of 100 meters to the overtaking vessel's projected course, and consequently the third spatial view being violated before `TS` has passed `OS` as required by the final spatial view. At this point, the requirement as formalised by the mTSC cannot be satisfied anymore and the monitor reports a violation. As soon as this happens the test run could be terminated and considered a fail. Beyond objective and automatic evaluation this may make test conduction more efficient by providing a criterion for early termination of test runs. In operation the same or a similar monitor could also be used to trigger a safety mechanism as soon as a violation is detected.

7 Conclusion

This work presents an integrated process for scenario-based testing of MASS, based on maritime Traffic Sequence Charts (mTSCs), a visual formalism adapted from the automotive domain to describe maritime scenarios. Applying mTSCs as a formal specification for abstract maritime traffic scenarios and their

corresponding requirements establishes a link across different process phases and provides an objective representation that can be evaluated automatically.

The overall process is structured around three core steps. First, hazardous operating scenarios are identified by means of a hazard analysis. Concrete scenarios for the SuT are then generated to be executed in a simulation. During the simulation, the behaviour of the SuT is evaluated against the mTSCs and their requirements. An online monitor provides live feedback on whether the requirements are being satisfied or violated during runtime.

As our approach formalises relevant scenarios for use in subsequent steps within a comprehensive process, this also enables scenarios to be reused in different test situations, for example after an update or when renewing a previous certification. Using mTSCs as an unambiguous representation of scenarios enables the generation and maintenance of consistent scenario catalogues, allowing for repeated evaluation in various test procedures and tools.

For the stakeholders involved, the presented procedure further enables comparability and allows the early provision of tests. For example, developers could use a test catalogue to prepare for the acceptance tests their systems have to pass. Other potential applications include training remote operation centre (ROC) operators and providing scenario monitors to offer Vessel Traffic Services (VTS) operators continuous situational awareness.

Acknowledgement
The work was conducted within the 'AMISIA - Advanced Port Maintenance: Intelligent, Sustainable and Automated Dredging' project, which is funded by the German Federal Ministry for Transport (BMV), in the IHATEC programme under grant ID 19H21003D and under the framework of the project "FuturePorts". FuturePorts started in January 2022 and is led by the Program Directorate Transport within the German Aerospace Center (DLR), whose support we greatly appreciate. We further thank our colleague Paula Wegerich who provided insight and expertise that greatly assisted the research.

References

[1] Hans-Christoph Burmeister, Jonathan Weisheit, Julius Kuechle, Luka-Franziska Bluhm, and Michael Bergmann. An emerging market? the european maritime industry's view on autonomous maritime systems: A survey. *Journal of Physics: Conference Series*, 2867(1):012017, 2024.

[2] Birte Kramer, Christian Neurohr, Matthias Büker, Eckard Böde, Martin Fränzle, and Werner Damm. Identification and Quantification of Hazardous Scenarios for Automated Driving. In Marc Zeller and Kai Höfig, editors, *Model-Based Safety and Assessment*, volume 12297, pages 163–178. Springer International Publishing, Cham, 2020.

[3] Anna Austel, Matthias Steidel, and Bernd Westphal. Formal specification of situations in scenario-based testing of maritime assistance systems. In *European Workshop on Maritime Systems Resilience and Security (MARESEC 2024)*, June 2024.

[4] W. Damm, S. Kemper, E. Möhlmann, T. Peikenkamp, and A. Rakow. Using traffic sequence charts for the development of HAVs. In *ERTS 2018*, 2018.

[5] Werner Damm, Eike Möhlmann, Thomas Peikenkamp, and Astrid Rakow. A Formal Semantics for Traffic Sequence Charts: Essays Dedicated to Edward A. Lee on the Occasion of His 60th Birthday. In *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, pages 182–205. July 2018.

[6] Xiang-Yu Zhou, Zheng-Jiang Liu, Feng-Wu Wang, Zhao-Lin Wu, and Ren-Da Cui. Towards applicability evaluation of hazard analysis methods for autonomous ships. *Ocean Engineering*, 214:107773, 2020.

[7] Zhihong Li, Di Zhang, Bing Han, and Chengpeng Wan. Risk and reliability analysis for maritime autonomous surface ship: A bibliometric review of literature from 2015 to 2022. *Accident Analysis & Prevention*, 187:107090, 2023.

[8] Chia-Hsun Chang, Christos Kontovas, Qing Yu, and Zaili Yang. Risk assessment of the operations of maritime autonomous surface ships. *Reliability Engineering & System Safety*, 207:107324, 2021.

[9] Bekir Sahin, Anis Yazidi, Dumitru Roman, and Ahmet Soylu. Ontology-based fault tree analysis algorithms in a fuzzy environment for autonomous ships. *IEEE Access*, 9:40915–40932, 2021.

[10] Børge Rokseth, Ingrid Bouwer Utne, and Jan Erik Vinnem. A systems approach to risk analysis of maritime operations. *Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability*, 231(1):53–68, 2017.

[11] Tom Arne Pedersen, Åse Neverlien, Jon Arne Glomsrud, Imran Ibrahim, Sigrid Marie Mo, Martin Rindarøy, Tobias Torben, and Børge Rokseth. Evolution of safety in marine systems: From system-theoretic process analysis to automated test scenario generation. *Journal of Physics: Conference Series*, 2311(1):012016, jul 2022.

[12] Tom Arne Pedersen, Chanjei Vasanthan, Kristian Karolius, Øystein Engelhardtsen, Koen Pieter Houweling, and Are Jørgensen. Generating structured set of encounters for verifying automated collision and grounding avoidance systems. *Journal of Physics: Conference Series*, 2618(1):012013, oct 2023.

[13] Tobias Rye Torben, Jon Arne Glomsrud, Tom Arne Pedersen, Ingrid B Utne, and Asgeir J Sørensen. Automatic simulation-based testing of autonomous ships using Gaussian processes and temporal logic. *Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability*, 237(2):293–313, 2023.

[14] CheeKuang Tam, Richard Bucknall, and Alistair Greig. Review of Collision Avoidance and Path Planning Methods for Ships in Close Range Encounters. *The Journal of Navigation*, 62(3):455–476, July 2009. Publisher: Cambridge University Press.

[15] Raphael Zaccone, Michele Martelli, and Massimo Figari. A COLREG-Compliant Ship Collision Avoidance Algorithm. In *2019 18th European Control Conference (ECC)*, pages 2530–2535, June 2019.

[16] L. P. Perera, J. P. Carvalho, and C. Guedes Soares. Intelligent Ocean Navigation and Fuzzy-Bayesian Decision/Action Formulation. *IEEE Journal of Oceanic Engineering*, 37(2):204–219, April 2012.

[17] M. Constapel, P. Koch, and H.-C. Burmeister. On the implementation of a rule-based system to perform assessment of colregs onboard maritime autonomous surface ships. In *The International Maritime and Port Technology and Development Conference (MTEC) & The 4th International Conference on Maritime Autonomous Surface Ships (ICMASS)*, volume 2311 of *Journal of Physics: Conference Series*, Singapore, April 2022.

[18] Ivan Porres, Sepinoud Azimi, Sébastien Lafond, Johan Lilius, Johanna Salokannel, and Mirva Salokorpi. On the verification and validation of ai navigation algorithms. In *Global Oceans 2020: Singapore – U.S. Gulf Coast*, pages 1–8, 2020.

[19] Ivan Porres, Sepinoud Azimi, and Johan Lilius. Scenario-based testing of a ship collision avoidance system. In *In*, pages 545–52. IEEE Computer Society, 2020.

[20] Maria Riveiro, Giuliana Pallotta, and Michele Vespe. Maritime anomaly detection: A review. *WIREs Data Mining and Knowledge Discovery*, 8(5):e1266, September 2018.

[21] Joao Lourenço, Joao Costa Seco, and Carla Ferreira. Monitoring of spatio-temporal properties with nonlinear SAT solvers. In *Formal Methods for Industrial Critical Systems: 27th International Conference, FMICS 2022, Warsaw, Poland, September 14–15, 2022, Proceedings*, volume 13487, page 155. Springer Nature, 2022.

[22] Srajan Goyal, Alberto Griggio, Jacob Kimblad, and Stefano Tonetta. Automatic generation of scenarios for system-level simulation-based verification of autonomous driving systems. *Electronic Proceedings in Theoretical Computer Science*, 395:113–129, November 2023.

[23] Kok Soon Oliver Tan and Sian Soo Tng. An integrated maritime reasoning and monitoring system. In *2012 15th International Conference on Information Fusion*, pages 1345–1350, 2012.

[24] International Maritime Organization (IMO). MSC.1-Circ.1638 - Outcome Of The Regulatory Scoping Exercise For The Use Of Maritime Autonomous Surface Ships (MASS), June 2021.

[25] Lokukaluge P. Perera. Deep Learning Toward Autonomous Ship Navigation and Possible COLREGs Failures. *Journal of Offshore Mechanics and Arctic Engineering*, 142(3), December 2019.

[26] David Reiher and Axel Hahn. Review on the current state of scenario-and simulation-based V&V in application for maritime traffic systems. In *OCEANS 2021: San Diego–Porto*, pages 1–9. IEEE, 2021.

[27] Kristine Bruun Ludvigsen. Open Simulation Platform - a collaborative effort to facilitate system integration, 2019.

[28] International Maritime Organization. *COLREG: Convention on the International Regulations for Preventing Collisions at Sea, 1972*. IMO Publication. International Maritime Organization, 2003.

[29] Christian Neurohr, Lukas Westhofen, Tjark Koopmann, Eike Möhlmann, Eckard Böde, and Axel Hahn. On Scenario Formalisms for Automated Driving, 2025.

[30] Anna Austel, Lukas Panneke, Janusz Piotrowski, Nina Wetzig, Matthias Steidel, and Bernd Westphal. Using monitoring of maritime traffic scenarios in the validation of maritime systems. In *2025 Symposium on Maritime Informatics and Robotics (MARIS)*, 2025 (To appear).

[31] Christian Neurohr, Lukas Westhofen, Martin Butz, Martin Herbert Bollmann, Ulrich Eberle, and Roland Galbas. Criticality analysis for the verification and validation of automated vehicles. *IEEE Access*, 9:18016–18041, 2021.

[32] Birte Kramer, Christian Neurohr, Matthias Büker, Eckard Böde, Martin Fränzle, and Werner Damm. Identification and quantification of hazardous scenarios for automated driving. In Marc Zeller and Kai Höfig, editors, *Model-Based Safety and Assessment*, pages 163–178, Cham, 2020. Springer International Publishing.

[33] Eckard Böde, Matthias Büker, Werner Damm, Martin Fränzle, Birte Kramer, Christian Neurohr, and Sebastian Vander Maelen. Identifikation und Quantifizierung von Automationsrisiken für hochautomatisierte Fahrfunktionen. page 75, October 2020.

[34] Leonardo De Moura and Nikolaj Bjørner. Satisfiability modulo theories: introduction and applications. *Communications of the ACM*, 54(9):69–77, 2011.

[35] Jan Becker, Tjark Koopmann, Birte Neurohr, Christian Neurohr, Lukas Westhofen, Boris Wirtz, Eckard Böde, and Werner Damm. Simulation of Abstract Scenarios: Towards Automated Tooling in Criticality Analysis. pages 42–51. February 2022.

[36] Jan Steffen Becker. Safe linear encoding of vehicle dynamics for the instantiation of abstract scenarios. In *International Conference on Formal Methods for Industrial Critical Systems*, pages 3–20. Springer, 2024.

[37] Zinchenko Serhii, Tovstokoryi Oleh, Nosov Pavlo, Popovych Ihor, and Kyrychenko Kostiantyn. Pivot point position determination and its use for manoeuvring a vessel. *Ships and offshore structures*, 18(3):358–364, 2023.

[38] Nikolaj Bjørner, Leonardo de Moura, Lev Nachmanson, and Christoph M. Wintersteiger. Programming z3. In Jonathan P. Bowen, Zhiming Liu, and Zili Zhang, editors, *Engineering Trustworthy Software Systems: 4th International School, SETSS 2018, Chongqing, China, April 7–12, 2018, Tutorial Lectures*, pages 148–201. Springer International Publishing, Cham, 2019.

[39] Comité International Radio-Maritime (CIRM). Route plan exchange format - RTZ. `https://cirm.org/rtz-xml-schemas`. Accessed: 2024-11-15.

[40] David Reiher and Axel Hahn. Ad hoc HLA simulation model derived from a model-based traffic scenario. *Simulation*, 99(8):859–882, 2023.

[41] N. Rüssmeier, A. Lamm, and A. Hahn. A generic testbed for simulation and physical-based testing of maritime cyber-physical system of systems. *Journal of Physics: Conference Series*, 1357(1):012025, October 2019.

[42] Ralf Stemmer, Ishan Saxena, Lukas Panneke, Dominik Grundt, Anna Austel, Eike Möhlmann, and Bernd Westphal. Runtime monitoring of complex scenario-based requirements for autonomous driving functions. *Science of Computer Programming*, 244:103301, 2025.